과제 #6

4190.414A 멀티코어 컴퓨팅(001)

Due: 2020년 6월 14일(월) 23시 59분

1 문제 1: CUDA를 이용한 multi-GPU 병렬화

행렬 곱셈을 수행하는 CUDA 예시 프로그램이 주어진다. CUDA의 사용을 위해서 `~/.bashrc`에 다음을 추가하자.

```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

다음은 실행 예시이다.

```
$ sbatch run.sh -v -n 10 512 512 512
$ Submitted batch job 14049
(...)
$ cat    slurm-14049.out
...
Using 4 devices
[GPU 0] GeForce RTX 2080
[GPU 1] GeForce RTX 2080
[GPU 2] GeForce RTX 2080
[GPU 3] GeForce RTX 2080
...
Calculating...(iter=9) 0.022737 sec
Validating...
Result: VALID
Avg. time: 0.024850 sec
Avg. throughput: 10.802144 GFLOPS
```

프로그램을 수정하여 성능을 높여보자. (Hint: OpenCL에서 했던 최적화를 그대로 적용하면 될 것이다.) 주의 사항은 다음과 같다.

- **1개의 실습 서버 계산노드에 탑재된 NVIDIA GeForce GTX 2080 GPU 4개를 사용한다.** 주어진 `run.sh` 스크립트를 그대로 사용하면 된다.
- `./run.sh` 의 첫 부분인 `SBATCH` 옵션을 바꾸어 GPU 개수를 설정할 수 있다. 주석처럼 보이지만 `sbatch` 커맨드에서 사용하는 옵션들이다. 일반적인 GPU 개수에 대해 잘 작동하도록 프로그램을 작성하는 것을 추천하지만, 채점은 single node & GPU 4개에 대해서만 진행할 것이다.
- **메모리 전송을 `mat_mul` 함수 내에서 수행하여야 한다.**
- 여러 스레드를 사용하는 것은 허용한다. OpenMP 의 사용도 허용한다. 단, CPU에서 행렬 곱의 일부를 수행하는 것은 금지한다.

- `mat_mul.cu`, `Makefile`만 수정할 수 있다. 다른 파일은 채점 시에 예시 코드로 덮어씌워진다.

보고서에는 다음 내용을 포함하여 작성한다.

- 병렬화 방법에 대한 설명
- OpenCL과 비교하여 CUDA 프로그래밍 경험 및 느낀 점 (상식적인 내용이 포함되어 있으면 감점하지 않음)

채점 기준은 다음과 같다.

보고서 (20%) 필요한 내용이 모두 포함되어 있으면 만점.

정확성 (40%) 4096 이하의 임의의 M, N, K에 대해서 `-v` 옵션을 통한 validation을 통과해야 한다.

성능 (40%) $M = N = K = 8192$ 옵션을 주고 실행했을 때, 1200 GFLOPS를 넘으면 만점. 그 이하는 비율에 따라 점수를 부여한다. 답이 틀린 경우 0점.

2 제출 방법

- 조교 메일(`jinpyo@aces.snu.ac.kr`)로 보고서를 포함한 모든 파일을 하나의 파일(e.g., .zip, .tar.gz)로 압축 후 첨부하여 제출한다.
- 메일 제목은 [mc2021] 계정이름_HW6으로 한다. (e.g., mc99_HW6)
- 첨부파일명은 계정이름_HW6.확장자으로 한다. (e.g., mc99_HW6.zip, mc99_HW6.tar.gz)
- 제출한 메일은 기계적으로 처리되므로 위의 내용을 지키지 않을 시 누락될 수 있으니 잘 지켜주시기 바랍니다.
- Grace day를 사용하고자 하는 경우에는 메일 내용에 이를 반드시 포함한다.